UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/591,680	07/24/2007	Martin C. Herbordt	BU-115XX	1481

207 7590 04/27/2011
WEINGARTEN, SCHURGIN, GAGNEBIN & LEBOVICI LLP
TEN POST OFFICE SQUARE
BOSTON, MA 02109

EXAMINER	
COYER, RYAN D	
ART UNIT	PAPER NUMBER
2197	
MAIL DATE	DELIVERY MODE
04/27/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/591,680	**Applicant(s)** HERBORDT ET AL.
	Examiner Ryan D. Coyer	**Art Unit** 2197

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 September 2006*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-60* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-60* is/are rejected.

7)☒ Claim(s) *7,15, 29, and 35* is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 September 2006* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *9/5/2006*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

This is in response to Application 10/591680, filed on 9/5/2006, in which claims 1-60 are presented for examination.

Specification

The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following amendment to the title is suggested: “System and Method for Programmable Logic Acceleration of Data Processing Applications and Compiler Therefor.”

Claim Objections

Claims 7, 15, 29 and 35 are objected to because of the following informalities: claims 15 and 29 do not end with a period (“."); claim 7 contains an extraneous semicolon (“;”) on the third line; claim 35 is a duplicate of claim 3. Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 21-24 and 31-34 are “single means” claims. A single means claim, wherein a means recitation does not appear in combination with another recited element of means, is subject to an undue breadth rejection under 35 U.S.C. 112, first paragraph. *In re Hyatt*, 708 F.2d 712, 714-715 (Fed. Cir. 1983). A single means claim covers every conceivable structure (means) for achieving the stated property (result) while the specification discloses at most only those known to the inventor. *See* MPEP 2164.08(a). Accordingly, the instant claims are rejected because they fail to comply with 35 U.S.C. 112, first paragraph.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 13, 16, 38-39, and 41-45 are method claims that recite "means for" language in a way that renders the claims ambiguous. More specifically, the claims' preambles recite "method . . . comprising the steps of", but at least one of the putative method steps is a “means for” limitation. The claims will be examined as if the words “means for” were excised. Furthermore, the claims each recite the limitation “identifying correlating available and needed for the coprocessor to provide application specific accelerate processing,” the meaning of which is unclear. The aforementioned limitation will be examined as if it were amended to recite “identifying resources available and needed for the coprocessor to provide application specific accelerate processing,” which is consistent with similar claims 30 and 31. Clarification is respectfully requested.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

<u>Claims 1-12, 14-15, 17-25, 27-29, 31-37, 40, 46-47, and 51</u> are rejected under 35 U.S.C. 102(b) as being anticipated by Ussery et al., USPAT 6,484,304, hereinafter "Ussery." Claims 21-25, 27-29, 31-34, 46-47, and 51 properly recite "means plus function" language and will be interpreted in accordance with 35 U.S.C. 112, sixth paragraph. Examiner asserts that the specification does not preclude the interpretations of claims 21-25, 27-29, 31-34, 46-47, and 51 employed in the following rejections.

<u>Regarding claim 1</u>, Ussery anticipates **"[a]n accelerated processor for use in massive data manipulations specific to an application comprising:**

a workstation having a general purpose processor and a coprocessor connection; (see, e.g., fig. 2 sec. 62, 64; col. 5 ln. 5-23; "The PSA analysis system 62 is a graphical analysis environment that allows the user to analyze certain characteristics of their application running on a target PSA IC"; "A PSA Image Loader/Programmer 64 loads the program image 60 into the PSA IC 54. This loads the binary program image into on-chip program memory on the PSA IC.")

an application specific coprocessor system at said connection; (see, e.g., fig. 2. sec. 54; col. 4 ln. 17-22, col. 5 ln. 20-23; "loads the program image 60 into the

PSA IC 54. This loads the binary program image into on-chip program memory on the PSA IC.")

said coprocessor system having programming code which is a assembled as instructions for said specific application in combination with accelerator environment specific requirements, independently provided." (see. e.g., fig. 2 sec. 52, 56, 58, 60; col. 4 ln. 23-30, 64-67; col. 5 ln. 1-4, 20-30; "The user program 56 containing the custom code that defines the system specification for the ASIC and the application libraries 58 is parsed by the PSA Compiler 52. The PSA compiler 52 converts the user program 56 into a program image 60 of the system specifications for the ASIC that comprises a series of microtasks. Each microtask is a Very Long Instruction Word (VLIW) program for a target task engine in the PSA IC 54.").

Regarding claim 2, Ussery anticipates **"[a]n application specific coprocessor system** (see, e.g., fig. 2. sec. 54; col. 4 ln. 17-22, col. 5 ln. 20-23; "loads the program image 60 into the PSA IC 54. This loads the binary program image into on-chip program memory on the PSA IC.") **for use with a processor for use in massive data manipulations specific to an application and adapted for attachment to a workstation having a general purpose processor,** (see, e.g., fig. 2 sec. 62, 64; col. 5 ln. 5-23; "The PSA analysis system 62 is a graphical analysis environment that allows the user to analyze certain characteristics of their application running on a target PSA IC"; "A PSA Image Loader/Programmer 64 loads the program image 60 into the PSA IC 54. This loads the binary program image into on-chip program memory on the PSA IC.") **said coprocessor system having programming code which is assembled as**

instructions for said specific application in combination with accelerator environment specific requirements, independently provided.” (see. e.g., fig. 2 sec. 52, 56, 58, 60; col. 4 ln. 23-30, 64-67; col. 5 ln. 1-4, 20-30; “The user program 56 containing the custom code that defines the system specification for the ASIC and the application libraries 58 is parsed by the PSA Compiler 52. The PSA compiler 52 converts the user program 56 into a program image 60 of the system specifications for the ASIC that comprises a series of microtasks. Each microtask is a Very Long Instruction Word (VLIW) program for a target task engine in the PSA IC 54.”).

Regarding claim 3, Ussery anticipates **“[t]he coprocessor of claim 2 wherein: said environment specific instructions are accessed by a compiler in response to user input in an application specific form.”** (see, e.g., fig. 3 and associated text; col. 4 ln. 64 – col. 5 ln. 4; “The PSA compiler 52 converts the user program 56 into a program image 60 of the system specifications for the ASIC that comprises a series of microtasks.”).

Regarding claim 4, Ussery anticipates **“[t]he coprocessor of claim 3 wherein said compiler comprises one or more of: user interface to permit an application trained non circuit design trained user to enter instructions to achieve accelerated performance,** (col. 5 ln. 5-20; “graphical analysis environment that allows the user to analyze certain characteristics of their application running on a target PSA IC.”; col. 4 ln. 55-58; “reduces the design cycle time and the level of skill required to produce the custom integrated circuit.”) **means to create an internal representation reflecting the operational characteristics of a coprocessor corresponding to**

application specific accelerated processing needs, (see, e.g., col. 5 ln. 5-20; "The PSA simulator is a software model of a specific configuration of the PSA IC hardware that can execute the program image 60 produced by the PSA compiler 52") **means for identifying bit demands for the application specific coprocessor acceleration function,** [NB: Ussery does not explicitly disclose this limitation, as set forth in the 103 rejections below, but Ussery need not do so to anticipate this claim] **mapper means for identifying resources available and needed for the coprocessor to provide application specific accelerated processing,** (see, e.g., col. 5 ln. 54-62, col. 6 ln. 4-14, col. 7 ln. 3-13; user-defined functions are mapped to target task engines and then optimized) **balancing means for identifying the step by step hardware needs of the coprocessor for the application specific acceleration."** (see, e.g., col. 6 ln. 4-14; "The global analyzer 106 determines the optimal task engines on which to execute the functionality of a given task. This is done by analyzing the required operations of the task (addition, multiplication, etc.) and matching those to a task engine that can optimally support the collection of operations required by the task.").

Regarding claim 5, Ussery anticipates **"[t]he coprocessor of claim 4 wherein said mapper means accepts as input domain-specific policy information, estimates of the amount of logic needed for each processing element, and hardware context information that states what amounts of each logic resource exist on a given coprocessor to enable the largest possible number of processing elements said coprocessor can support."** (see, e.g., col. 5 ln. 54-62, col. 6 ln. 4-14, col. 7 ln. 3-13; user-defined functions are mapped to target task engines and then optimized).

Regarding claim 6, Ussery anticipates **“[t]he coprocessor of claim 5 wherein said balancing means analyzes the processing speed of said coprocessor at each step and allocates parallel hardware in proportion to a speed requirement.”** (see, e.g., col. 7 ln. 3-13; “The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.”).

Regarding claim 7, Ussery anticipates **“[t]he coprocessor of claim 3, wherein said compiler further includes one or more of prerecorded information reflecting the programming requirements for a general area of applications; programming content which reflects application requirements and hardware characteristics; and coprocessor specific hardware availability.”** (see, e.g., col. 6 ln. 4-14; application requirements are evaluated and mapped to appropriate task engines comprising the necessary processing power.).

Regarding claim 8, Ussery anticipates **“[a] method for programming an accelerating coprocessor** (see, e.g., fig. 2. sec. 54; col. 4 ln. 17-22, col. 5 ln. 20-23; “loads the program image 60 into the PSA IC 54. This loads the binary program image into on-chip program memory on the PSA IC.”) **comprising the steps of: accessing data reflective of programming requirements for a general area of applications.”** (see, e.g,. fig. 2 and associated text; col. 4 ln. 23-30; “To generate the ASIC using the method of the present invention, an end user (not shown) develops a user program 56 that includes custom code which defines the system specification for the ASIC. The user program 56 may incorporate application libraries 58 of verified code that perform certain predetermined functions likely to be found in target application. These application libraries 58 are essentially virtual intellectual property ("Virtual IP") blocks.”).

Regarding claim 9, Ussery anticipates **“[t]he method for programming an accelerating coprocessor of claim 8 comprising the steps of: accessing data reflective of programming content which reflects application requirements and hardware characteristics.”** (see, e.g., fig. 3 and associated text; col. 4 ln. 64 – col. 5 ln. 4; “The PSA compiler 52 converts the user program 56 into a program image 60 of the system specifications for the ASIC that comprises a series of microtasks.”).

Regarding claim 10, Ussery anticipates **“[t]he method for programming an accelerating coprocessor of claim 8 comprising the steps of: accessing data reflective of coprocessor specific hardware availability.”** (see, e.g., col. 7 ln. 3-13; (see, e.g., col. 7 ln. 3-13; “The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.”).

Regarding claim 11, Ussery anticipates **“[t]he method of claim 8 further comprising the steps of: permitting an application trained non circuit design trained user to enter instructions to achieve accelerated performance.”** (see, e.g., col. 4 ln. 56-63; col. 5 ln. 5-20; “graphical analysis environment that allows the user to analyze certain characteristics of their application running on a target PSA IC.”; col. 4 ln. 55-58; “reduces the design cycle time and the level of skill required to produce the custom integrated circuit.”).

Regarding claim 12, Ussery anticipates **“[t]he method of claim 8 further comprising the steps of: creating an internal representation reflecting the operational characteristics of a coprocessor corresponding to application specific accelerated processing needs.”** (see, e.g., col. 5 ln. 5-20; “The PSA simulator is a software model of a specific configuration of the PSA IC hardware that can execute the program image 60 produced by the PSA compiler 52”).

Regarding claim 14, Ussery anticipates **“[t]he method of claim 8 further comprising the steps of: identifying the step by step hardware needs of the coprocessor for the application specific acceleration.”** (see, e.g., col. 6 ln. 4-14; “The global analyzer 106 determines the optimal task engines on which to execute the functionality of a given task. This is done by analyzing the required operations of the task (addition, multiplication, etc.) and matching those to a task engine that can optimally support the collection of operations required by the task.”).

Regarding claim 15, Ussery anticipates **“[a] method of compiling data for programming an accelerating coprocessor** (see, e.g., fig. 2. sec. 54; col. 4 ln. 17-22, col. 5 ln. 20-23; “loads the program image 60 into the PSA IC 54. This loads the binary program image into on-chip program memory on the PSA IC.”) **comprising the steps of: permitting an application trained non circuit design trained user to enter**

instructions to achieve accelerated performance.” (see, e.g., col. 4 ln. 56-63; col. 5 ln. 5-20; “graphical analysis environment that allows the user to analyze certain characteristics of their application running on a target PSA IC.”; col. 4 ln. 55-58; “reduces the design cycle time and the level of skill required to produce the custom integrated circuit.”).

Regarding claim 17, Ussery anticipates **“[a] method of compiling data for programming an accelerating coprocessor** (see, e.g., fig. 2. sec. 54; col. 4 ln. 17-22, col. 5 ln. 20-23; “loads the program image 60 into the PSA IC 54. This loads the binary program image into on-chip program memory on the PSA IC.”) **comprising the steps of: identifying resources available and needed for the coprocessor to provide application specific accelerated processing.”** (see, e.g., col. 6 ln. 4-14; “The global analyzer 106 determines the optimal task engines on which to execute the functionality of a given task. This is done by analyzing the required operations of the task (addition, multiplication, etc.) and matching those to a task engine that can optimally support the collection of operations required by the task.”).

Regarding claim 18, Ussery anticipates **“[t]he method of claim 17 wherein said identifying step further includes the step of accepting as input domain-specific policy information, estimates of an amount of logic needed for each processing element, and hardware context information that states what amounts of each logic resource exist in a given coprocessor and providing a design maximizing a number of processing elements that the coprocessor can support.”** (see, e.g., col. 5 ln. 54-62, col. 6 ln. 4-14, col. 7 ln. 3-13; user-defined functions are mapped to target task engines and then optimized).

Regarding claim 19, Ussery anticipates **"[a] method of compiling data for programming an accelerating coprocessor comprising the steps of: identifying the step by step hardware needs of the coprocessor for the application specific acceleration."** (see, e.g., col. 6 ln. 4-14; "The global analyzer 106 determines the optimal task engines on which to execute the functionality of a given task. This is done by analyzing the required operations of the task (addition, multiplication, etc.) and matching those to a task engine that can optimally support the collection of operations required by the task.").

Regarding claim 20, Ussery anticipates **"[t]he method of claim 19 wherein said identifying step includes analyzing the processing speed at each step and allocating parallel hardware in proportion to a processing speed requirement."** (see, e.g., col. 7 ln. 3-13; "The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.").

Regarding claim 27, Ussery anticipates **"[t]he compiler of claim 21 further comprising means for identifying resources available and needed for the coprocessor to provide application specific accelerated processing."** (see, e.g., col. 6 ln. 4-14; "The global analyzer 106 determines the optimal task engines on which to execute the functionality of a given task. This is done by analyzing the required operations of the task (addition, multiplication, etc.) and matching those to a task engine that can optimally support the collection of operations required by the task.").

Regarding claim 35, Ussery anticipates **"[t]he coprocessor of claim 2 wherein: said environment specific instructions are accessed by a compiler in response to user input in an application specific form."** (see, e.g., fig. 3 and associated text; col. 4 ln. 64 – col. 5 ln. 4; "The PSA compiler 52 converts the user program 56 into a program image 60 of the system specifications for the ASIC that comprises a series of microtasks.").

Regarding claim 36, Ussery anticipates **"[t]he method of claim l0 further comprising the steps of: permitting an application trained non circuit design trained user to enter instructions to achieve accelerated performance."** (see, e.g., col. 4 ln. 56-63; col. 5 ln. 5-20; "graphical analysis environment that allows the user to analyze certain characteristics of their application running on a target PSA IC."; col. 4 ln. 55-58; "reduces the design cycle time and the level of skill required to produce the custom integrated circuit.").

Regarding claim 37, Ussery anticipates **"[t]he method of claim 36 further comprising the steps of: creating an internal representation reflecting the operational characteristics of a coprocessor corresponding to application specific accelerated processing needs."** (see, e.g., col. 5 ln. 5-20; "The PSA simulator is a software model of a specific configuration of the PSA IC hardware that can execute the program image 60 produced by the PSA compiler 52").

Regarding claim 40, Ussery anticipates **"[t]he method of claim 11 further**

comprising the steps of: creating an internal representation reflecting the operational characteristics of a coprocessor corresponding to application specific accelerated processing needs." (see, e.g., col. 5 ln. 5-20; "The PSA simulator is a software model of a specific configuration of the PSA IC hardware that can execute the program image 60 produced by the PSA compiler 52").

Regarding claims 21-25, 28-29, 31-34, 46-47, and 51, the scope of the instant claims does not differ substantially from that of claims 8-12, 14-15, 17-20, 36-37, and 40. Accordingly, the rejections of claims 8-12 apply, *mutatis mutandis*, to claims 21-25; the rejections of claims 14-15 apply, *mutatis mutandis*, to claims 28-29; the rejections of claims 17-20 apply, *mutatis mutandis*, to claims 31-34; the rejections of claim 36-37 apply, *mutatis mutandis*, to claims 46-47; and the rejection of claim 40 applies, *mutatis mutandis*, to claim 51.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.

2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 13, 16, 26, 30, 38-39, 41-45, 48-50, and 52-60 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ussery in view of Shackleford et al., USPAT 5,896,521, hereinafter “Shackleford.” Claims 26, 30, 48-50, and 52-60 properly recite "means plus function" language and will be interpreted in accordance with 35 U.S.C. 112, sixth paragraph. Examiner asserts that the specification does not preclude the interpretations of claims 26, 30, 48-50, and 52-60 employed in the following rejections.

Regarding claim 13, Ussery discloses **“[t]he method of claim 8 further comprising the steps of: identifying resources available and needed for the coprocessor to provide application specific accelerated processing.”** (see, e.g., col. 5 ln. 54-62, col. 6 ln. 4-14, col. 7 ln. 3-13; user-defined functions are mapped to target task engines and then optimized; “The global analyzer 106 determines the optimal task engines on which to execute the functionality of a given task. This is done by analyzing the required operations of the task (addition, multiplication, etc.) and matching those to a task engine that can optimally support the collection of operations required by the task.”).

Ussery does not explicitly disclose the limitation further comprising “**identifying bit demands for the application specific coprocessor acceleration function.”** Shackleford discloses a “processor synthesis system . . . which can reduce the number of software development steps specifically adapted to the processor architecture to be synthesized” (col. 3, ln. 47-51) wherein “the CPU bit width is customized to the requirement of [an] application." (col. 6 ln. 50-53). In other words, Shackleford identifies the bit width required for an application with a view toward synthesizing an application specific integrated circuit (ASIC) that is optimized for the application.

Ussery and Shackleford are directed toward the fields of processor synthesis (also known as generating ASICs) and therefore are analogous art. At the time of the invention, it would have appeared obvious to one of ordinary skill in the art to enhance the processor synthesis method of Ussery by incorporating the bit-width analysis method of Shackleford. A source of motiviation would have been to ensure that "maximum cost efficiency is obtained." (Shackleford, col. 6 ln. 52-53). The result of combining Shackleford and Ussery would be both predictable and useful: the minimum processor bit-width would be used, thereby reducing development costs. Therefore, the instant claim is unpatentable over Ussery in view of Shackleford.

Regarding claim 16, Ussery discloses **"[a] method of compiling data for programming an accelerating coprocessor** (see, e.g., fig. 2. sec. 54; col. 4 ln. 17-22, col. 5 ln. 20-23; "loads the program image 60 into the PSA IC 54. This loads the binary program image into on-chip program memory on the PSA IC.") **comprising the steps of: creating an internal representation reflecting the operational characteristics of a coprocessor corresponding to application specific accelerated processing needs."** (see, e.g., col. 5 ln. 5-20; "The PSA simulator is a software model of a specific configuration of the PSA IC hardware that can execute the program image 60 produced by the PSA compiler 52").

Ussery does not explicitly disclose the limitation further comprising "**identifying bit demands for the application specific coprocessor acceleration function.**" Shackleford discloses a "processor synthesis system . . . which can reduce the number of software development steps specifically adapted to the processor architecture to be synthesized" (col. 3, ln. 47-51) wherein "the CPU bit width is customized to the requirement of [an] application." (col. 6 ln. 50-53). In other words, Shackleford identifies the bit width required for an application with a view toward synthesizing an application specific integrated circuit (ASIC) that is optimized for the application.

Ussery and Shackleford are directed toward the fields of processor synthesis (also known as generating ASICs) and therefore are analogous art. At the time of the invention, it would have appeared obvious to one of ordinary skill in the art to enhance the processor synthesis method of Ussery by incorporating the bit-width analysis method of Shackleford. A source of motiviation would have been to ensure that "maximum cost efficiency is obtained." (Shackleford, col. 6 ln. 52-53). The result of combining Shackleford and Ussery would be both predictable and useful: the minimum processor bit-width would be used, thereby reducing development costs. Therefore, the instant claim is unpatentable over Ussery in view of Shackleford.

Regarding claim 26, Ussery discloses **"[t]he compiler of claim 21."** Ussery does not explicitly disclose the limitation "**further comprising: means for identifying bit demands for the application specific coprocessor acceleration function.**" Shackleford discloses a "processor synthesis system . . . which can reduce the number of software development steps specifically adapted to the processor architecture to be synthesized" (col. 3, ln. 47-51) wherein "the CPU bit width is customized to the requirement of [an] application." (col. 6 ln. 50-53). In other words, Shackleford identifies the bit width required for an application with a view toward synthesizing an application specific integrated circuit (ASIC) that is optimized for the application.

Ussery and Shackleford are directed toward the fields of processor synthesis (also known as generating ASICs) and therefore are analogous art. At the time of the invention, it would have appeared obvious to one of ordinary skill in the art to enhance the processor synthesis method of Ussery by incorporating the bit-width analysis method of Shackleford. A source of motiviation would have been to ensure that "maximum cost efficiency is obtained." (Shackleford, col. 6 ln. 52-53). The result of combining Shackleford and Ussery would be both predictable and useful: the minimum processor bit-width would be used, thereby reducing development costs. Therefore, the instant claim is unpatentable over Ussery in view of Shackleford.

Regarding claim 30, the scope of the instant claim does not differ substantially from that of claim 16. Accordingly, the rejection of claim 16 applies, *mutatis mutandis*, to the instant claim.

Regarding claim 38, Ussery discloses **"[t]he method of claim 37 further comprising the steps of: identifying resources available and needed for the coprocessor to provide application specific accelerated processing."** (see, e.g., col. 5 ln. 54-62, col. 6 ln. 4-14, col. 7 ln. 3-13; user-defined functions are mapped to target task engines and then optimized; "The global analyzer 106 determines the optimal task engines on which to execute the functionality of a given task. This is done by analyzing the required operations of the task (addition, multiplication, etc.) and matching those to a task engine that can optimally support the collection of operations required by the task.").

Ussery does not explicitly disclose the limitation further comprising "**identifying bit demands for the application specific coprocessor acceleration function.**" Shackleford discloses a "processor synthesis system . . . which can reduce the number of software development steps specifically adapted to the processor architecture to be synthesized" (col. 3, ln. 47-51) wherein "the CPU bit width is customized to the requirement of [an] application." (col. 6 ln. 50-53). In other words, Shackleford identifies the bit width required for an application with a view toward synthesizing an application specific integrated circuit (ASIC) that is optimized for the application.

Ussery and Shackleford are directed toward the fields of processor synthesis (also known as generating ASICs) and therefore are analogous art. At the time of the invention, it would have appeared obvious to one of ordinary skill in the art to enhance the processor synthesis method of Ussery by incorporating the bit-width analysis method of Shackleford. A source of motiviation would have been to ensure that "maximum cost efficiency is obtained." (Shackleford, col. 6 ln. 52-53). The result of combining Shackleford and Ussery would be both predictable and useful: the minimum processor bit-width would be used, thereby reducing development costs. Therefore, the instant claim is unpatentable over Ussery in view of Shackleford.

Regarding claim 39, Ussery discloses **"[t]he method of claim 38 further comprising the steps of: identifying the step by step hardware needs of the**

coprocessor for the application specific acceleration.” (see, e.g., col. 7 ln. 3-13; “The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.”).

Regarding claim 41, Ussery discloses **“[t]he method of claim 40 further comprising the steps of: identifying resources available and needed for the coprocessor to provide application specific accelerated processing.”** (see, e.g., col. 5 ln. 54-62, col. 6 ln. 4-14, col. 7 ln. 3-13; user-defined functions are mapped to target task engines and then optimized; “The global analyzer 106 determines the optimal task engines on which to execute the functionality of a given task. This is done by analyzing the required operations of the task (addition, multiplication, etc.) and matching those to a task engine that can optimally support the collection of operations required by the task.”).

Ussery does not explicitly disclose the limitation further comprising “**identifying bit demands for the application specific coprocessor acceleration function.”** Shackleford discloses a “processor synthesis system . . . which can reduce the number of software development steps specifically adapted to the processor architecture to be synthesized” (col. 3, ln. 47-51) wherein “the CPU bit width is customized to the requirement of [an] application.” (col. 6 ln. 50-53). In other words, Shackleford identifies the bit width required for an application with a view toward synthesizing an application specific integrated circuit (ASIC) that is optimized for the application.

Ussery and Shackleford are directed toward the fields of processor synthesis (also known as generating ASICs) and therefore are analogous art. At the time of the invention, it would have appeared obvious to one of ordinary skill in the art to enhance the processor synthesis method of Ussery by incorporating the bit-width analysis method of Shackleford. A source of motiviation would have been to ensure that "maximum cost efficiency is obtained." (Shackleford, col. 6 ln. 52-53). The result of combining Shackleford and Ussery would be both predictable and useful: the minimum processor bit-width would be used, thereby reducing development costs. Therefore, the instant claim is unpatentable over Ussery in view of Shackleford.

Regarding claim 42, Ussery discloses **"[t]he method of claim 41 further comprising the steps of: identifying the step by step hardware needs of the coprocessor for the application specific acceleration."** (see, e.g., col. 7 ln. 3-13; "The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.").

Regarding claim 43, Ussery discloses **"[t]he method of claim 12 further comprising the steps of: identifying resources available and needed for the coprocessor to provide application specific accelerated processing."** (see, e.g., col. 5 ln. 54-62, col. 6 ln. 4-14, col. 7 ln. 3-13; user-defined functions are mapped to target task engines and then optimized; “The global analyzer 106 determines the optimal task engines on which to execute the functionality of a given task. This is done by analyzing the required operations of the task (addition, multiplication, etc.) and matching those to a task engine that can optimally support the collection of operations required by the task.”).

Ussery does not explicitly disclose the limitation further comprising “**identifying bit demands for the application specific coprocessor acceleration function.”** Shackleford discloses a “processor synthesis system . . . which can reduce the number of software development steps specifically adapted to the processor architecture to be synthesized” (col. 3, ln. 47-51) wherein “the CPU bit width is customized to the requirement of [an] application.” (col. 6 ln. 50-53). In other words, Shackleford identifies the bit width required for an application with a view toward synthesizing an application specific integrated circuit (ASIC) that is optimized for the application.

Ussery and Shackleford are directed toward the fields of processor synthesis (also known as generating ASICs) and therefore are analogous art. At the time of the invention, it would have appeared obvious to one of ordinary skill in the art to enhance the processor synthesis method of Ussery by incorporating the bit-width analysis method of Shackleford. A source of motiviation would have been to ensure that “maximum cost efficiency is obtained.” (Shackleford, col. 6 ln. 52-53). The result of combining Shackleford and Ussery would be both predictable and useful: the minimum processor bit-width would be used, thereby reducing development costs. Therefore, the instant claim is unpatentable over Ussery in view of Shackleford.

Regarding claim 44, Ussery discloses **"[t]he method of claim 43 further comprising the steps of: identifying the step by step hardware needs of the coprocessor for the application specific acceleration."** (see, e.g., col. 7 ln. 3-13; "The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.").

Regarding claim 45, Ussery discloses **"[t]he method of claim 13 further comprising the steps of: identifying the step by step hardware needs of the coprocessor for the application specific acceleration."** (see, e.g., col. 7 ln. 3-13; "The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.").

Regarding claims 48-50 and 52-57, the scope of the instant claims does not differ substantially from that of claims 38-39 and 41-43. Accordingly, the rejection of claim 38 applies, *mutatis mutandis*, to claims 48-49; the rejection of claim 39 applies, *mutatis mutandis*, to claim 50; the rejection of claim 41 applies, *mutatis mutandis*, to claims 52 and 53; the rejection of claim 42 applies, *mutatis mutandis*, to claims 54 and 57; and the rejection of claim 43 applies, *mutatis mutandis*, to claims 55 and 56.

Regarding claim 58, Ussery discloses **"[t]he compiler of claim 26 further comprising means for identifying resources available and needed for the coprocessor to provide application specific accelerated processing.**" (see, e.g., col. 7 ln. 3-13; "The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.").

Regarding claim 59, Ussery discloses **"[t]he compiler of claim 58 further comprising: means for identifying the step by step hardware needs of the coprocessor for the application specific acceleration."** (see, e.g., col. 7 ln. 3-13; "The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.").

Regarding claim 60, Ussery discloses **"[t]he compiler of claim 27 further comprising: means for identifying the step by step hardware needs of the coprocessor for the application specific acceleration."** (see, e.g., col. 7 ln. 3-13; "The microtask analyzer and code generator 110 optimizes the microtasks for parallelism if multiple computation units are in the target task engine. Rather than executing each operation in sequence, the microtask analyzer and code generator 110 determines operations which can be executed independent of each other and allocates these operations to different computation units.”).

Conclusion

The prior art made of record on form PTO-892, 'Notice of References Cited', but not relied upon in the above rejections, is considered pertinent to applicant’s disclosure. The aforementioned prior art addresses subject matter disclosed in the specification but not necessarily presented in the instant claims.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to RYAN D. COYER, whose telephone number is (571) 270-5306, and whose fax number is (571) 270-6306. The examiner normally may be reached via phone on Mon-Thurs, 9a-8p. If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Li B. Zhen, can be reached on (571) 272-3768. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ryan D. Coyer/
Examiner, Art Unit 2197

/CHAMELI C. DAS/
Primary Examiner, Art Unit 2197
Dated: 4/25/11